Complexity aspects of web services composition

Karima Ennaoui, Lhouari Nourine, Farouk Toumani

Complexity aspects of web services composition

Karima Ennaoui, Lhouari Nourine, and Farouk Toumani

LIMOS, CNRS, Universite Blaise Pascal
73 rue des landais, AUBIERE, France

Abstract. The web service composition problem can be stated as follows: given a finite state machine M, representing a service business protocol, and a set of finite state machines $\mathcal{R}$, representing the business protocols of existing services, the question is to check whether there is a simulation relation between M and the shuffle product closure of $\mathcal{R}$. In fact the shuffle product is a subclass of the communication free petri net and basic parallel processes, for which the same problem of simulation is known to be 2-Exptime-hard.
This paper studies the impact of several parameters on the complexity of this problem. We show that the problem is *Exptime-complete* if we bound either: *(i)* the number of instances of services in $\mathcal{R}$ that can be used in a composition, or *(ii)* the number of the so-called hybrid states in the finite state machines of $\mathcal{R}$. Moreover, for the particular case where the bound of the hybrid states of the finite state machines of $\mathcal{R}$ is set to 2, we show that the composition problem is in 3-*Exptime*.

1 Introduction

Web Services [1] is a new computing paradigm that tends to become a technology of choice to facilitate interoperation among autonomous and distributed applications. The UDDI consortium defines Web services as *self-contained, modular business applications that have open, Internet-oriented, standards-based interfaces*[1]. Several models have been proposed in the literature to describe different facets of services. In particular, the importance of specifying external behaviour of services, also called service business protocols, has been highlighted in several research works [4,2,3]. Through literature, different models have been used to represent web service business protocols. The Finite States Machines (FSM) formalism is widely adopted in this context to model *statefull* applications exposed as web services where states represent the different phases that a service may go through while transitions represent "abstract" activities that a service can perform [4,2,3].

We consider in this paper the problem of Web Service Composition (WSC). This problem arises from the situation where none of the existing services can provide a requested functionality. In this case, the idea is to find out, algorithmically, if the target functionality could be composed out of the existing services (components repository). This automatic approach of composition simplifies the

[1] http://www.uddi.org/

development of software by reusing existing components and offers capabilities to customize complex systems built on the fly [7]. We focus more particularly on a specific instance of WSC, namely the (business) protocol synthesis problem, which can be stated as follows: given a set of business protocols of available services and given a business protocol of a target service, is it possible to synthesize automatically a mediator that *implements* the target service using the existing ones?

[12] shows that when business protocols are described by means of FSMs, the WSC problem can then be formalized as the problem of deciding whether there exists a simulation relation between the target protocol and the shuffle (or asynchronous product) of the available ones. This result is however based on the implicit assumption that at most one *instance* of each available service can be used in a composition. This setting has been extended in [7] to the case where the number of instances that can be used in a composition is unbounded. WSC is formalized in this latter case as a simulation problem between an FSM and an infinite state machine, called Product Closure State Machine (PCSM), that is able to compute the shuffle closure of an FSM.

Shuffle product of FSMs (and PCSM) is a subclass of Basic Parallel Processes (BPP) [5] , the class of communication free petri nets: every transition has at most one input place. Simulation of FSM by BPP was proven Expspace-hard by Lasota [11] and 2-Exptime-hard in [6].

Complexity analysis of WSC was first considered by Musholl *et al.*[12], under the aforementioned implicit assumption, where it is shown Exptime-Complete. In case of unbounded instances, the WSC problem has been proved decidable with an Ackermanian function as upper bound in [7]. The proof of [7] is based on Dickson lemma, and hence cannot be exploited to derive tighter upper bounds. An Expspace-hard lower bound is given by Lasota[11]. The source of complexity derived from the analysis of the algorithm given in [7] is related to the presence of the so-called hybrid states[2] in the components and loops in the target: if the target FSM is loop free, the WSC problem becomes NP-complete and when the components are hybrid state free the problem is proven Exptime.

In this paper, we consider additional parameters related to bounded/unbounded web services composition. We consider as inputs an FSM M (the target protocol) and a set of FSMs $\mathcal{R}$ (the protocols of the available services) and we investigate the complexity of testing simulation between M and the shuffle closure of $\mathcal{R}$, represented as a PCSM [7]. More precisely, we study the complexity of the following problems:

1. $WSC(M, \mathcal{R})$: The problem of composing M using an unbounded number of instances of $\mathcal{R}$.
2. $BC(M, \mathcal{R}, k)$: The problem of composing M using at most k instances of each FSM in $\mathcal{R}$.
3. $UCHS(M, \mathcal{R}, k)$: The problem of composing M using an unbounded number of instances of $\mathcal{R}$, with the number of hybrid states in $\mathcal{R}$ is bounded by k.

[2] Hybrid states of an FSM are final states with outgoing transitions and correspond to unbounded places in Petri net terminology.

Table 1 displays known and new complexity results regarding the WSC problem.

M	Acyclic FSM	general FSM
$BC(M, \mathcal{R}, 1)$	NP-complete[7]	Exptime-complete [12]
$BC(M, \mathcal{R}, k)$	NP-complete[7]	Exptime-complete (this paper)
$WSC(M, \mathcal{R})$	NP-complete [7]	Decidable [7]
$UCHS(M, \mathcal{R}, 0)$	NP-complete[7]	Exptime-complete (this paper)
$UCHS(M, \mathcal{R}, 1)$	NP-complete[7]	Exptime-complete (this paper)
$UCHS(M, \mathcal{R}, 2)$	NP-complete[7]	3-Exptime (this paper)

Table 1: Complexity of Web services composition problem.

Paper organisation Section 2 recalls some basic definitions needed in this paper. In section 2, investigates the problem of bounded web services composition and proves that it is Exptime-Complete. In section 2, we consider the web service composition when the number of hybrid states is bounded. We show that this problem is Exptime-Complete for $k = 0$, $k = 1$ and 3-Exptime for $k = 2$. We conclude in section 5.

2 Preliminaries

Finite State Machine We consider in this paper service business protocols formally described as FSMs. We recall below the definition of such machines.

Definition 1. ***(Finite State Machine (FSM))***
A State Machine (SM) M is a tuple $M = (\Sigma_M, Q_M, F_M, q_M^0, \delta_M)$, where: Σ_M is a finite alphabet, Q_M is a set of states, $\delta_M \subseteq Q_M \times \Sigma_M \times Q_M$ is a set of labeled transitions, $F_M \subseteq Q_M$ is a set of final states, and $q_M^0 \in Q_M$ is the initial state. If Q_M is finite then M is called a Finite State Machine (FSM).

Moreover, a state $q \in Q_M$ is called: **accessible**, if there exists a path from the initial state to q; **co-accessible**, if there exists a path from q to a final state; **intermediate**, if $q \notin F_M$ and $\exists p_1, p_2 \in Q_M$, $s.t$ $(p_1, a, q) \in \delta_M$ and $(q, b, p_2) \in \delta_M$, we denote by $I(M)$ the set of intermediate states of M; **hybrid**, if $q \in F_M$, $q \neq q_0$ and there exist at least one transition $(q, b, p) \in \delta_M$, with $p \in Q_M$ and $b \in \Sigma$, the set of hybrid states is denoted $H(M)$ and **terminal**, if $q \in F_M$ and is not hybrid.

We consider here only FSMs where all states are both accessible and co-accessible. We define the **norm of a state** q as the finite length of the shortest path from q to a final state. **The norm of an FSM** M, noted *norm(M)*, is the maximal norm of its states.

k-Iterated Product Machine (k-IPM) and Product State Machine (PCSM) We start by defining the shuffle (asynchronous product) and union operations on FSMs:

Definition 2. ***(Asynchronous product and Union of two FSMs)*** *Let $M = (\Sigma_M, Q_M, F_M, q^0_M, \delta_M)$ and $M' = (\Sigma_{M'}, Q_{M'}, F_{M'}, q^0_{M'}, \delta_{M'})$ be two FSMs. We have :*

- *The* ***shuffle or asynchronous product*** *of M and M', denoted $M \times M'$, is an FSM $(\Sigma_M \cup \Sigma_{M'}, Q_M \times Q_{M'}, F_M \times F_{M'}, (q^0_M, q^0_{M'}), \lambda)$ where the transition function λ is defined as follows: $\lambda = \{((q, q'), a, (q_1, q_1')) : ((q, a, q_1) \in \delta_M$ and $q' = q'_1)$ or $((q', a, q_1') \in \delta_{M'}$ and $q = q_1)\}$.*
- *The* ***union*** *of M and M', denoted $M \cup M'$, is the FSM $(\Sigma_M \cup \Sigma_{M'} \cup \{\epsilon\}$, $Q_M \cup Q_{M'} \cup \{q_0\}$, $F_M \cup F_{M'}$, q_0, $\delta_M \cup \delta_{M'} \cup \{(q_0, \epsilon, q^0_M), (q_0, \epsilon, q^0_{M'})\})$.*

For a set of available FSMs $\mathcal{R} = \{M_1, ...M_i\}$, we consider a compact structure that abstracts all possible executions that can be produced using the components of $\mathcal{R}$. First, we begin by the simple case where each M_j can be used only once:

Definition 3. ***(Asynchronous product of FSMs set)*** *The asynchronous product of all the subsets elements of FSMs repository $\mathcal{R} = \{M_1....M_m\}$ is the FSM: $\odot(\mathcal{R}) = \bigcup_{\{M_{i_1},...,M_{i_j}\} \subseteq \mathcal{R}} (M_{i_1} \times ... \times M_{i_j})$ where $j \in [0, i]$.*

Second, we consider the case where the number of copies of each $M_j \in \mathcal{R}$ is bounded by an integer k:

Definition 4. ***(k-iterated product of FSMs set $\mathcal{R}$)*** *The* ***k-iterated product of*** *$\mathcal{R}$ is defined by $\mathcal{R}^{\otimes_k} = \mathcal{R}^{\otimes_{k-1}} \times \odot(\mathcal{R})$ with $\mathcal{R}^{\otimes_1} = \odot(\mathcal{R})$.*

Finally, we consider the general case where the number of instances of each $M_j \in \mathcal{R}$ is unbounded. This corresponds to the product closure of $\mathcal{R}$ [7]:

Definition 5. ***(Product closure of FSMs set)*** *The* ***product closure*** *of $\mathcal{R}$, noted $\mathcal{R}^{\otimes}$, is defined as: $\mathcal{R}^{\otimes} = \bigcup_{i=0}^{+\infty} \mathcal{R}^{\otimes_i}$.*

The **P**roduct **C**losure **M**achine (**PCSM**) of $\mathcal{R}$, defined in [7] and proven equivalent to $\mathcal{R}^{\otimes}$, is the SM $(\Sigma_{\mathcal{R}}, \mathcal{C}_{\mathcal{R}^{\otimes}}, F_{\mathcal{C}}, c_0, \Phi_{\mathcal{R}^{\otimes}})$, where:

1. $\Sigma_{\mathcal{R}} = \bigcup_{M_j \in \mathcal{R}} \Sigma_{M_j}$;
2. $\mathcal{C}_{\mathcal{R}^{\otimes}}$ is the set of states (also called configurations of $\mathcal{R}^{\otimes}$). $\mathcal{C}_{\mathcal{R}^{\otimes}} \subset \mathbb{N}^n$, with: $n = n_I(\mathcal{R}) + n_H(\mathcal{R})$ with: $n_I(\mathcal{R}) = \Sigma_{M_j \in \mathcal{R}} |I(M_j)|$ and $n_H(\mathcal{R}) = \Sigma_{M_j \in \mathcal{R}} |H(M_j)|$. For each configuration c, $c[m]$ (the m^{th} component of c) is called a witness of the unique state $q_m \in Q_{M_j}$. Note that:
   - q_m is an intermediate state, if $1 \leq m \leq n_I(\mathcal{R})$;
   - q_m is an hybrid state, if $n_I(\mathcal{R}) + 1 \leq m \leq n$.

   In an abuse of notation, we use $c[m]$ and $c[q_m]$ interchangeably.
3. $F_{\mathcal{C}}$ is the set of final states. $F_{\mathcal{C}} = \{c \in \mathcal{C}_{\mathcal{R}^{\otimes}} | c[m] = 0,$ *for each:* $1 \leq m \leq n_I(\mathcal{R})\}$;

4. $c_0 = \{0\}^n$ is the initial state of $\mathcal{R}^{\otimes}$;
5. $\Phi_{\mathcal{R}^{\otimes}} \subseteq \mathcal{C}_{\mathcal{R}^{\otimes}} \times \Sigma_{\mathcal{R}} \times \mathcal{C}_{\mathcal{R}^{\otimes}}$ is the set of transitions. we have $(c_1, a, c_2) \in \Phi_{\mathcal{R}^{\otimes}}$ **iff**:
   - there exists $(q_0, a, q) \in Q_{M_j}$, such that: q_0 is the initial state of M_j and $c_2[q] = c_1[q] + 1$ and $c_2[p'] = c_1[p']$ for each $p' \neq q$.
   - there exists $(p, a, q) \in Q_{M_j}$, such that: $c_2[p] = c_1[p] - 1$, $c_2[q] = c_1[q] + 1$ and $c_2[p'] = c_1[p']$ for each $p' \neq p, q$.
   - there exists $(p, a, q) \in Q_{M_j}$, such that: q is a final state or the initial state, $c_2[p] = c_1[p] - 1$ and $c_2[p'] = c_1[p']$ for each $p' \neq p$.

Simulation preorder We recall below the definition of the simulation preorder between two SMs.

Definition 6. ***(Simulation)***
Let $M = (\Sigma_M, Q_M, F_M, q_M^0, \delta_M)$ and $N = (\Sigma_M, Q_M, F_M, q_M^0, \delta_M)$ be two SMs. A state $p \in Q_M$ is simulated by a state $q \in Q_N$, denoted $p{<<}_{(M,N)}\ q$ ($p{<<}q$ when M and N are understood from context), iff the following two conditions hold:

1. *$\forall a \in \Sigma_M$ and $\forall p' \in Q_M$ such that $(p, a, p') \in \delta_M$, there exists $(q, a, q') \in \delta_N$ such that $p'{<<}q'$, and*
2. *if $p \in F_M$, then $q \in F_N$.*

M is simulated by N, denoted $M{<<}N$, ***iff*** *the initial state of N simulates the initial sate of M.*

Observe that, by definition, each transition of a PCSM can at most increase or decrease a configuration component by 1. In addition, if a configuration is final then all intermediate states witnesses are equal to 0. Therefore, given a set of FSMs $\mathcal{R}$ and $c \in \mathcal{C}_{\mathcal{R}^{\otimes}}$, we have $\Sigma_{q \in \bigcup_{M_i \in \mathcal{R}} I(M_i)} c[q] \leq norm(c)$. Moreover, since final states can only be simulated by final ones, then for M an FSM and $p \in Q_M$, if $p{<<}c$ then $norm(c) \leq norm(p)$. Hence, we are able to derive the following property.

Property 1. **(Intermediate witnesses bound) [7]** For $c \in \mathcal{C}_{\mathcal{R}^{\otimes}}$ and $p \in Q_M$, if $p \in c$ then $\Sigma_{q \in \bigcup_{M_i \in \mathcal{R}} I(M_i)} c[q] \leq norm(p)$. we denote $\mathcal{C}^M_{\mathcal{R}^{\otimes}} = \{c \in \mathcal{C}_{\mathcal{R}^{\otimes}} | \Sigma_{q \in \bigcup_{M_i \in \mathcal{R}} I(M_i)} c[q] \leq norm(M)\}$.

In [7], the WSC problem in the unbounded case is reduced to simulation test between an FSM and a PCSM and this later problem is proved to be decidable. The proof of the termination of the algorithm given in [7] is based on the following property:

Property 2. **(configuration cover) [7]** Let c and c' be two configurations of $\mathcal{R}^{\otimes}$, such that: $c[m] = c'[m]$, $m \in [1, I]$ *and* $c[m] \leq c'[m]$, $m \in]I, I+H]$. if $q{<<}c$, where q is a state of a SM M, then $q{<<}c'$.

we say that c' covers c, denoted $c \lhd c'$.

We introduce below the algorithm of [7], focusing the presentation on the structure of its execution tree.

Definition 7. ***(Simulation Tree)***
We call a simulation tree $T_{sim}(M, \mathcal{R}^{\otimes}) = (V, v_0, E)$ *with:* $v_0 = (q_M^0, c_0)$ *is the root of the tree;* $V \subset Q_M \times \mathcal{C}_{\mathcal{R}^{\otimes}}^M$ *is the set of nodes; If* $(q, c) \in V$ *and* q *is final in* M *then so is* c *in* $\mathcal{R}^{\otimes}$*;* $E \subset V \times V$ *is the set of the tree's edges.* $\forall e = ((p, c), (q, d)) \in E : \exists a \in \Sigma_M$ *s.t* $(p, a, q) \in \delta_M$ *and* $(c, a, d) \in \Phi_{\mathcal{R}^{\otimes}}$*.*
$v = (p, c) \in V$ *is a leaf in* $T_{sim}(M, \mathcal{R}^{\otimes})$ *iff* p *is terminal in* A *or there exists* $(p, c') \in P = \{v_0 ... v\}$ *such that* $c \triangleleft c'$ *where* P *is the set of ancestors of* v *in* $T_{sim}(M, \mathcal{R}^{\otimes})$*.*

In the next section, we shall bound the size of this tree in the case of bounded WSC problem (i.e., when the instances of services allowed to be used in the simulation is bounded by a parameter k).

3 Bounded Composition

We call a *bounded* WSC problem, a service composition problem where the number of copies of each web service in the repository $\mathcal{R}$ used to compose the target M is bounded a priori by an integer k. This problem is formally stated as follows.

Problem 1. **Bounded Composition** $BC(M, \mathcal{R}, k)$
Input : $\mathcal{R}$ a set of FSMs; M a target FSM; k an integer.
Question : $M << \mathcal{R}^{\otimes_k}$?

The particular case $BC(M, \mathcal{R}, 1)$ has been investigated by Muscholl and Walukiewicz [12] where it is shown to be Exptime-Complete. We shall prove in this section that $BC(M, \mathcal{R}, k)$ is also Exptime-Complete. We point out that the straightforward reduction of $BC(M, \mathcal{R}, k)$ to $BC(M, \mathcal{R}, 1)$, obtained by duplicating k times each service of $\mathcal{R}$, is not polynomial in the input size, since k may be large, and hence cannot be used to achieve our goal.

The parameter k drops the infinite aspect and reduces the search space. In this case, a loop in M can only be simulated by loops in $\mathcal{R}$. For example, one can observe that, in figure 1, S_t is not simulated by $\{R_1, R_3\}^{\otimes^k}$ for every $k \in \mathbb{N}$. This is because when we repeat the loop in S_t $(k+1)$ times, there is no corresponding execution in $\{R_1, R_3\}^{\otimes^k}$. However, we have $S_t << \{R_1, R_2\}^{\otimes^k}$, for any $k \geq 1$.

In the following, we give an upper bound of the number of states that might appear in $\mathcal{R}^{\otimes_k}$, with $k \in \mathbb{N}$.

Lemma 1. *Let* $\mathcal{R}$ *be a set of FSM and* k *is an integer. The number of states in* $\mathcal{R}^{\otimes_k}$ *is bounded by* $| \mathcal{C}_{\mathcal{R}^{\otimes_k}} | \leq | \{c \in \mathbb{N}^n \mid c[i] \leq k, i \in [1, n]\} | = O(2^{nlogk})$*, where* $n = |\mathcal{R}| + n_I(\mathcal{R}) + n_H(\mathcal{R})$*.*

This lemma reduces the search space to an exponential size and leads to the following theorem.

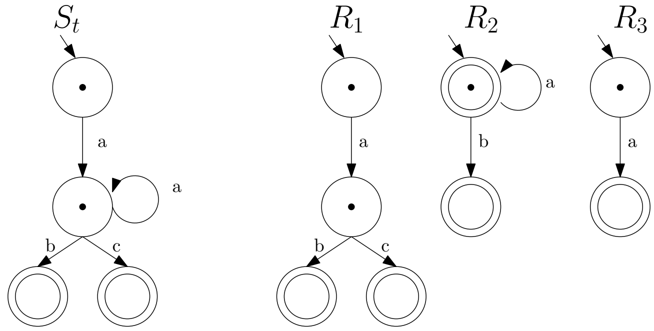

Fig. 1. A yes instance of $BC(M, \mathcal{R}, k)$ with $k = 1$.

Theorem 1. *$BC(M, \mathcal{R}, k)$ is Exptime-Complete*

Proof. `Exptime.` To show that $BC(M, \mathcal{R}, k)$ is Exptime, we bound the size of the simulation tree. A node of the simulation tree corresponds to (q, c) where q is a state of M and c a configuration of $\mathcal{R}^{\otimes_k}$. According to Lemma 1, the number of PCSM's configurations is bounded by k^n. So the number of nodes in the simulation tree is at most $|Q_M| \times k^n = 2^{nlog(k)+log(|Q_M|)}$ and therefore the complexity is in Exptime.

`Exptime-Hardness.` It can be deducted directly from the Exptime-Hardness of the particular case $BC(M, \mathcal{R}, 1)$ [12].

Another factor of complexity of the WSC problem is the number of hybrid states in the available services. We investigate next the effect of this parameter on the complexity of the WSC problem.

4 Bounded number of hybrid states

The presence of hybrid states is a source of complexity in a WSC problem. As mentioned before, the size of intermediate states witnesses in configurations of $\mathcal{R}^{\otimes}$ used to simulate M is bounded by $norm(M)$. We are however unable to provide a similar bound for the number of hybrid states witnesses. Figure 2 illustrates the different roles that an hybrid state of $\mathcal{R}$ can play to simulate a state of M. Indeed an hybrid state of $\mathcal{R}$, can be used as: (i) a terminal state, e.g., when testing wether $q_5 << (1, 1)$, we can consider the second hybrid state of $\mathcal{R}$ as a terminal state and terminate the test, or an intermediate state, e.g., when testing wether $q_2 << (1, 1)$, the second hybrid state of $\mathcal{R}$ here plays the role of intermediate state, or both a terminal and an intermediate state, e.g., when testing wether $q_1 << (1, 0)$, a transition of $\Phi_{\mathcal{R}^{\otimes}}$ labeled by $(b, (-1, 0))$ only appears in one branch in the simulation tree $\mathcal{T}_{sim}(M, \mathcal{R}^{\otimes})$. Hence, the first hybrid state of $\mathcal{R}^{\otimes}$ is considered intermediate in one branch and terminal in the other, or a hybrid state, e.g., when it is used to simulate an hybrid state of $H(M)$.

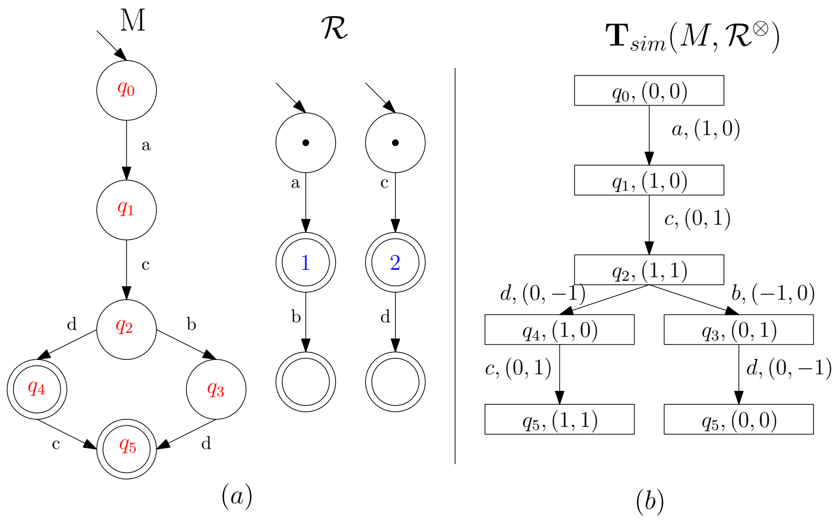

Fig. 2. Example of the simulation tree

We consider in the following the problem defined below.

Problem 2. **Unbounded Composition With limited number of Hybrid States** $UCHS(M, \mathcal{R}, k)$

Input : k an integer; $\mathcal{R}$ a set of FSMs, containing at most k hybrid states; M a target FSM.
Question : $M << \mathcal{R}^{\otimes}$?

It is worth noting that $UCHS(M, \mathcal{R}, k+1)$ is harder then $UCHS(M, \mathcal{R}, k)$. In the sequel, we progressively investigate the complexity of $UCHS(M, \mathcal{R}, k)$ problem for $k = 0$, then for $k = 1$ and finally for $k = 2$.

4.1 Case of composition without hybrid states (i.e. $k = 0$)

In this section, we are interested by the problem $UCHS(M, \mathcal{R}, 0)$. We first give a polynomial transformation, denoted $\mathcal{K}$, which is used to reduce $BC(M, \mathcal{R}, 1)$ to $UCHS(N, \mathcal{R}', 0)$. This transformation provides a mean to bound the number of instances used to prove simulation.

Definition 8. ***Transformation*** $\mathcal{K}$**.** *For an FSM* $M = (\Sigma_M, Q_M, F_M, q_0^M, \delta_M)$ *and a set of FSMs* $\mathcal{R} = \{M_1, ..., M_m\}$*, we define* $\mathcal{K}(M, \mathcal{R}) = (N, \mathcal{R}' = \{N_1, .., N_m\})$ *where:*

1. *Each* N_i *is built based on* M_i*, by adding a letter* t_i *to its alphabet, a final state* f_i *and a transition set* $\{(q_0^{M_i}, t_i, f_i)\} \cup \{(q, t_i, f_i) | q \in F_{M_i}\}$*. All final states of* M_i *become intermediate in* N_i*.*

2. *N is defined as:*
   – $\Sigma_N = \Sigma_M \cup \{t_i | 1 \leq i \leq m\}$;
   – $Q_N = Q_M \cup \{r_i | 1 \leq i \leq m\}$;
   – $F_N = \{r_m\}$;
   – $\delta_N = \delta_M \cup \{(q, t_1, r_1) | q \in F_M\} \cup \{(r_i, t_{i+1}, r_{i+1}) | 1 \leq i < m\}$.

Figure 3 shows an example of a transformation $\mathcal{K}$.

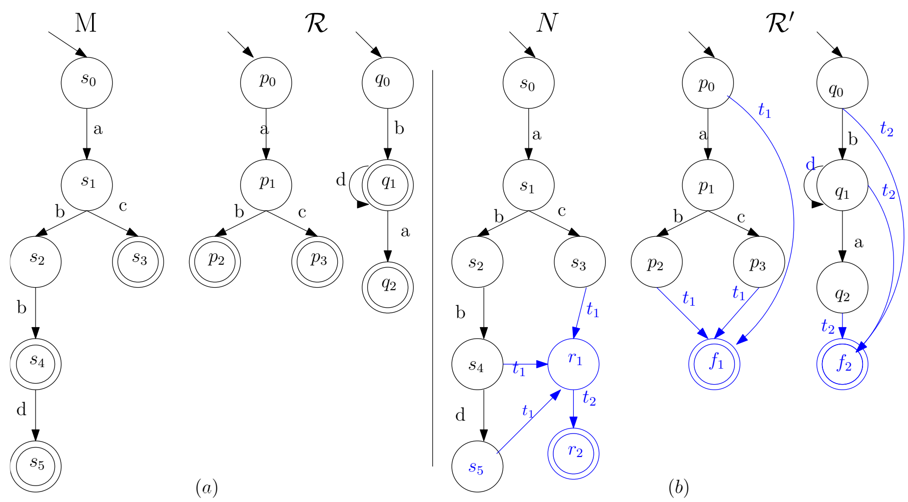

Fig. 3. *An example of transformation* $\mathcal{K}$

The following propositions show that the transformation $\mathcal{K}$ preserves the simulation preorder.

Proposition 1. *Let M be an FSM, $\mathcal{R} = \{M_1, ..., M_m\}$ be a set of FSMs and $\mathcal{K}(M, \mathcal{R}) = (N, \mathcal{R}' = \{N_1, .., N_m\})$. For p and q two states of respectively M and $\mathcal{R}^{\otimes_1}$, we have: $p << _{(M,(\mathcal{R})^{\otimes_1})} q$ iff $p << _{(N,(\mathcal{R}')^{\otimes_1})} q$.*

Proof. By construction of $\mathcal{K}(M, \mathcal{R})$, *if* $p << _{(M,\mathcal{R}^{\otimes_1})} q$ and p is terminal in M *then* $p << _{(N,(\mathcal{R}')^{\otimes_1})} q$.

We suppose next that:

If $(p, a, p') \in \delta_M$, $(q, a, q') \in \delta_{\mathcal{R}^{\otimes_1}}$ and $p' << _{(M,\mathcal{R}^{\otimes_1})} q'$, *then* $p' << _{(N,(\mathcal{R}')^{\otimes_1})} q'$.

and prove that $p << _{(N,(\mathcal{R}')^{\otimes_1})} q$.

For each $(p, a, p') \in \delta_N$, we have:

- *if* $a \in \Sigma_M$, *then* there exists $(q, a, q') \in \delta_{\mathcal{R}^{\otimes_1}} \subseteq \delta_{(\mathcal{R}')^{\otimes_1}}$ such that $p' << _{(N,(\mathcal{R}')^{\otimes_1})} q'$.
- else $a = t_1$, $p' = r_1$ and q is a product of final states of $\mathcal{R}$. therefore, there exists $(q, t_1, q') \in \delta_{(\mathcal{R}')^{\otimes_1}}$ such that $q' = (f_1, {q'}_{i_1}, ..., {q'}_{i_l})$ where ${q'}_{i_j}$ is final in $\mathcal{R}$ such that $p' << _{(N,(\mathcal{R}')^{\otimes_1})} q'$.

We conclude that if $p << _{(M,\mathcal{R}^{\otimes_1})} q$ then $p << _{(N,(\mathcal{R}')^{\otimes_1})} q$.

Reciprocally, we have $(p, a, p') \in \delta_N$ (respectively $\delta_{(\mathcal{R}')^{\otimes_1}}$) and $a \notin \{t_i | 1 \leq i \leq m\}$ iff $(p, a, p') \in \delta_M$ (respectively $\delta_{\mathcal{R}^{\otimes_1}}$). In addition, the definition of $\mathcal{K}$ ensures that if p is final in M and $p << _{(N,(\mathcal{R}')^{\otimes_1})} q$ then q is final in $\mathcal{R}^{\otimes_1}$. Hence if $p << _{(N,(\mathcal{R}')^{\otimes_1})} q$ then $p << _{(M,\mathcal{R}^{\otimes_1})} q$.

In particular, we take p as the initial state of M and q the initial state of $\mathcal{R}^{\otimes_1}$. This implies that:

Proposition 2. *Let M be an FSM, $\mathcal{R} = \{M_1, ..., M_m\}$ be a set of FSMs and $\mathcal{K}(M, \mathcal{R}) = (N, \mathcal{R}' = \{N_1, .., N_m\})$. We have: $M << \mathcal{R}^{\otimes_1}$ iff $N << (\mathcal{R}')^{\otimes}$.*

Proof. We have $N << (\mathcal{R}')^{\otimes_1}$ *iff* $N << (\mathcal{R}')^{\otimes}$. Indeed, each path that starts from the initial state to a final one in N contains exactly one transition labeled by t_i, for each $i \in [1, m]$ and a similar path in each N_i contains exactly one transition labeled by t_i.

Hence, $\mathcal{K}$ is a polynomial reduction of $BC(M, \mathcal{R}, 1)$ problem to the UCHS problem. This enables to derive the following result.

Theorem 2. *$UCHS(M, \mathcal{R}, 0)$ problem is Exptime-complete.*

Proof. According to proposition 2, the $\mathcal{K}$ transformation reduces $BC(M, \mathcal{R}, 1)$ to $UCHS(M, \mathcal{R}, 0)$ in polynomial time. Thus $UCHS(M, \mathcal{R}, 0)$ is Exptime-hard. Since it is also proven Exptime in [7], then $UCHS(M, \mathcal{R}, 0)$ is Exptime-complete.

4.2 Case of composition with one hybrid state

We consider the problem $UCHS(M, \mathcal{R}, 1)$ where M is an FSM and $\mathcal{R}$ a set of FSMs containing at most one hybrid state ($n_H(\mathcal{R}) \leq 1$). We denote $k_0 = |Q_A|.2^{n_I(\mathcal{R}).log(norm(M))}$. Two nodes (q, c) and (q', c') in a simulation tree are called comparable if $q = q'$ and either $c \lhd c'$ or $c' \lhd c$. The nodes (q, c) and (q', c') are said incomparable otherwise.

Property 3. Let $\mathcal{R}$ be a set of FSMs containing at most one hybrid state. Two configurations of $\mathcal{R}^{\otimes}$ are comparable by the cover relation, *iff* they have exactly the same intermediate witnesses.

Property 4. Let S be a set of nodes of $\mathcal{T}_{sim}(M, \mathcal{R}^{\otimes})$ that are pairwise incomparable, *then* $|S| \leq k_0$.

Proof. In configurations considered in $\mathcal{T}_{sim}(M, \mathcal{R}^{\otimes})$, intermediate witnesses are bounded by $norm(M)$ (property 1). Therefore and according to property 3, the number of incomparable configurations considered in $\mathcal{T}_{sim}(M, \mathcal{R}^{\otimes})$ is at most $2^{n_I(\mathcal{R}).log(norm(M))}$. Since $S \subset Q_M \times \mathcal{C}_{\mathcal{R}^{\otimes}}$, then $|S| \leq k_0$.

Proposition 3. *If $n_H(\mathcal{R}) \leq 1$, then foreach $(q, c) \in \mathcal{T}_{sim}(M, \mathcal{R}^{\otimes})$, $c[I+1] \leq {k_0}^2$.*

Proof. let P be a path in $\mathcal{T}_{sim}(M, \mathcal{R}^{\otimes})$ and $S = (v_n = (q_n, c_n))_{n \in Int \subset \mathbb{N}}$ be a sequence of nodes in P such that:
-v_i is the i^{th} node met in P that is comparable to one of its predecessors $v = (q_i, c)$; and
-For each $i, j \in Int$, v_i and v_j are incomparable.

If $Int = \emptyset$, then all nodes of P are not comparable. The size of P is then bounded by k_0, therefore, $c[I+1] \leq k_0$ for each (q,c) in P.

We suppose next that $I \neq \emptyset$ and take $Int = [1, k]$, $k \in \mathbb{N}$. We prove recursively that for each $n \in [1, k]$, $c_n[I+1] \leq n.k_0$.

For $n = 1$, we have $c_1[I+1] \leq k_0$.
For $1 < n < k$, we suppose that $c_n[I+1] \leq n.k_0$. Each node $v = (q, c)$ between v_n and v_{n+1} in P is either:

1. comparable to a node v_i with $i \in [1, n]$. In this case, $c[I+1] < c_i[I+1] \leq n.k_0$ (otherwise v should be a leaf).
2. incomparable to all its predecessors. The number of such nodes is bounded by k_0. And since transitions displacements is in $\{-1, 0, 1\}^{I+1}$, then we have $c[I+1] < n.k_0 + k_0$.

Therefore $c_{n+1}[I+1] \leq (n+1).K_0$.

Once we reach v_k, all its possible successors are comparable to a node v_i with $c[I+1] < c_i[I+1]$, except for the last one that is the leaf of P.

Finally, since $k < k_0$ (because S is a sequence of incomparable nodes), we conclude that each node of P is in $Q_A \times ([1, norm(A)]^I \times [1, {k_0}^2])$.

Since deciding simulation only requires to visit a node once, we argue next that this problem is in APspace: the size of a position of the simulation tree is polynomial in the input size (Proposition 3). Hence a polynomial space alternating turing machine can solve this simulation problem: universal states correspond to the target's and existential states correspond to the shuffle product's configurations. Given the above, we conclude that:

Lemma 2. *$UCHS(M, \mathcal{R}, 1)$ is in Exptime.*

To prove the Exptime-hardness of the problem, we recall that $UCHS(M, \mathcal{R}, 0)$ is Exptime-hard (theorem 2) and that $UCHS(M, \mathcal{R}, 1)$ is harder than $UCHS(M, \mathcal{R}, 0)$.

Theorem 3. *$UCHS(M, \mathcal{R}, 1)$ is Exptime-complete.*

4.3 Case of composition with two hybrid states

In this section, we consider the problem of unbounded composition of web services with at most 2 hybrid states in $\mathcal{R}$, *i.e.* $UCHS(M, \mathcal{R}, 2)$. Our approach is based on relating this simulation problem to the reachability issue.

For $x \in \mathbb{N}^{i_1}$ and $y \in \mathbb{N}^{i_2}$, we denote the concatenation of two vectors x and y, $(x.y) \in \mathbb{N}^{i_1+i_2}$ such that:

$$(x.y)[j] = \begin{cases} x[j] \ if \ j \in [1, i_1] \\ y[j] \ if \ j \in [i_1+1, i_1+i_2] \end{cases}$$

We define the 2-dimension Vector Addition System with States (VASS) [8] $\mathcal{V}_{M,\mathcal{R}}$ as follows:

- States: $S \subseteq Q_M \times \{c \in \mathbb{N}^{n_I(\mathcal{R})} | \sum_{i=1}^{i=n_I(\mathcal{R})} c[i] \leq norm(M)\}$.
- Transitions: $W \subseteq S \times S \times \{-1, 0, 1\}^2$ such that: $((q, c), (p, d), x) \in W$ *iff* there exists $a \in \Sigma_M$, $y \in \mathbb{N}^2$ such that $(p, a, q) \in Q_M$ *and* $((c, y), a, (d, y + x)) \in \Phi_{\mathcal{R}^\otimes}$ and $\sum_{i=1}^{i=n_I(\mathcal{R})} c[i] \leq norm(M)$ and $\sum_{i=1}^{i=n_I(\mathcal{R})} d[i] \leq norm(M)$.
- Initial configuration: the system starts with the state $(q_0^M, \{0\}^{n_I(\mathcal{R})})$ and the vector $(0, 0)$.

Figure 4 depicts an example of a VASS associated to an FSM M and a set of FSMs $\mathcal{R}$.

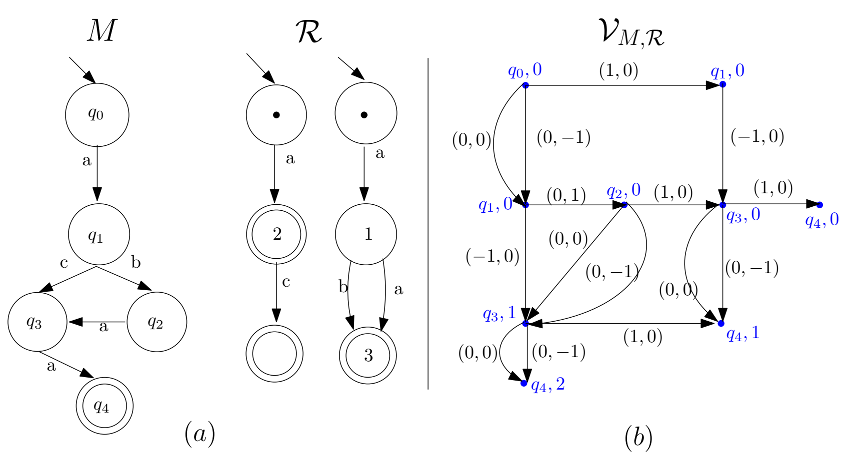

Fig. 4. *An example of a VASS associated to an FSM M and an FSMs set.*

The reachability issue in 2-dimension VASSs has been investigated by Hopcroft and Pansiot [8] in the general case where displacements are in $\mathbb{N}^2$. [8] gives an algorithm to prove the semi-linearity of the reachability set of such systems. The algorithm builds a tree T_{reach} labeled by 3-tuples (p, c, A_c) where p is the current state, $c \in \mathbb{N}^2$ is a vector reached in the system and $A_c \subset \mathbb{N}^2$. (p, c, A_c) denotes that every vector in the linear set $\{c + \alpha_1 a_1 + ... + \alpha_n a_n | i \in [1, n], a_i \in A_c \textit{ and } \alpha_i \in \mathbb{N}\}$ can be reached in state p from the initial configuration.

We consider in the following a simulation tree $T_{sim}(M, \mathcal{R}^{\otimes}) = (V, v_0, E)$, a recheability tree $T_{reach}(\mathcal{V}_{M,\mathcal{R}}) = (V', v'_0, E')$ and a function π defined as follows:

$$\begin{array}{rccc} \pi : & V' & \to & V \\ & ((p, c), x, A_x) & \mapsto & (p, (c.x)) \end{array}$$

The following proposition enables to establish a connection between paths in a simulation tree and a corresponding recheability tree.

Proposition 4. *Let $\mu = v_0 ... v_t$ be a path in $T_{sim}(M, \mathcal{R}^{\otimes})$. Then there exists a path $\mu' = v'_0 ... v'_t$ in $T_{reach}(\mathcal{V}_{M,\mathcal{R}})$ such that $v_i = \pi(v'_i)$, $i \in [0, t]$.*

Proof. We proof by induction on the length i of the path $\mu = v_0 ... v_t$.

For $i = 0$ we have $v_0 = (q_0^M, c_0) = \pi((q_0^M, \{0\}^{n_I(\mathcal{R})}), x_0, A_{c_0}) = \pi(v'_0)$.

Now suppose that the property is true for $i < t$ and $v_0 ... v_{i+1}$ is a path in $T_{sim}(M, \mathcal{R}^{\otimes})$. Then by hypothesis there exists a path $v'_0 ... v'_i$ in $T_{reach}(\mathcal{V}_{M,\mathcal{R}})$, such that $v_j = \pi(v'_j)$, $j \in [0, i]$.

Suppose that v'_i is a leaf in $T_{reach}(\mathcal{V}_{M,\mathcal{R}})$. Then according to the algorithm of Hopcroft and Pansiot [8], we have either:

- There exist $j \in [0, i-1]$ such that $v'_j = ((p, c), y, A_x)$ and $v'_i = ((p, c), x, A_x)$ with $y \leq x$ (see Algorithm **??**, line 1). This implies that $v_j \triangleleft v_i$, which contradicts that v_i is not a leaf in $T_{sim}(M, \mathcal{R}^{\otimes})$, i.e. $(c, y) \triangleleft (c, x)$.
- There is no transition from v'_i in the system (see Algorithm **??**, line 1). But for $v_i = (p, (c, x))$ and $v_{i+1} = (q, (d, y))$ we have $v_i v_{i+1} \in E$ which means that $(p, a, q) \in \delta_M$ and $((c, x), a, (d, y)) \in \Phi_{\mathcal{R}^{\otimes}}$. This implies that $((p, c), (q, d), y - x) \in W$. Contradiction.

Therefore, we have : $v'_{i+1} = ((q, d), y, A_y)$ is a successor of v'_i in $T_{reach}(\mathcal{V}_{M,\mathcal{R}})$, with $v_{i+1} = (q, (d, y))$. We conclude that μ' is a path in $T_{reach}(\mathcal{V}_{M,\mathcal{R}})$.

The following corollary is a consequence of Proposition 4.

Corollary 1. *$T_{sim}(M, \mathcal{R}^{\otimes})$ is a sub-tree of $T_{reach}(\mathcal{V}_{M,\mathcal{R}})$.*

Clearly the time complexity for computing $T_{sim}(M, \mathcal{R}^{\otimes})$ is dominated by the complexity of computing $T_{reach}(\mathcal{V}_{M,\mathcal{R}})$. Moreover we know from [9] that the size of $T_{reach}(\mathcal{V}_{M,\mathcal{R}})$ is in 2-Exptime. Hence, we derive the following complexity result.

Theorem 4. *$UCHS(M, \mathcal{R}, 2)$ is in 3-Exptime.*

Proof. According to [9], the size of $T_{reach}(\mathcal{V}_{M,\mathcal{R}})$ is of order $O(2^{2^{\alpha}})$ where $\alpha = max(|S|, |W|) \leq c \times (|Q_M| \times Norm(M)^{n_I(\mathcal{R})})^2$ with c is a constant. Then according to Corollary 1, the size of $T_{sim}(M, \mathcal{R}^{\otimes})$ is bounded by $2^{2^{2^{c_1 + c_2 * \beta}}}$ where c_1 and c_2 are constants and $\beta = log(|Q_M|) + n_I(\mathcal{R}) \times log(Norm(M))$.

Our proof for Theorem 4 can be seen more as an embedding of the search space explored by a simulation test to the one explored when the reachability issue is considered. This is an approach that can not so far be generalized because the best upper bound provided for vector addition systems reachability is non-primitive recursive; in fact even the existence of a primitive upper-bound is still open [10].

5 Conclusion

In this paper we have considered two parameters that are source of complexity of the web services composition problem. We have shown that among the considered problems, several instances remain Exptime-complete when a parameter is bounded. It remains an open question to identify whether the unbounded web services composition problem is in XP, where XP is the complexity class that contains all problems solvable in $O(n^{f(k)})$ with n is the size of M and $\mathcal{R}$, and f is a recursive function. Since $FPT \subset XP$, it is interesting as well to investigate what kind of fixed parameters can lead to FPT complexity.

References

1. Gustavo Alonso, Fabio Casati, Harumi Kuno, and Vijay Machiraju. *Web Services: Concepts, Architectures and Applications*. Springer, 2010.
2. B. Benatallah, F. Casati, and F. Toumani. Web Service Conversation Modeling: A Cornerstone for E-Business Automation. *IEEE Internet Computing*, 08(1):46–54, 2004.
3. D. Berardi, D.Calvanese, G. De Giacomo, R. Hull, and M. Mecella. Automatic composition of transition-based semantic web services with messaging. In *VLDB*, pages 613–624, 2005.
4. T. Bultan, X. Fu, R. Hull, and J. Su. Conversation specification: a new approach to design and analysis of e-service composition. In *WWW'03*. ACM, 2003.
5. Søren Christensen. Decidability and decomposition in process algebras. 1993.
6. Jean-Baptiste Courtois and Sylvain Schmitz. Alternating vector addition systems with states. In *Mathematical Foundations of Computer Science 2014*, pages 220–231. Springer, 2014.
7. Ramy Ragab Hassen, Lhouari Nourine, and Farouk Toumani. Protocol-based web service composition. In *ICSOC*, pages 38–53, 2008.
8. John Hopcroft and Jean-Jacques Pansiot. On the reachability problem for 5-dimensional vector addition systems. *TCS*, 8(2):135–159, 1979.
9. Rodney R Howell, Louis E Rosier, Dung T Huynh, and Hsu-Chun Yen. Some complexity bounds for problems concerning finite and 2-dimensional vector addition systems with states. *TCS*, 46:107–140, 1986.
10. S Rao Kosaraju. Decidability of reachability in vector addition systems (preliminary version). In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 267–281. ACM, 1982.
11. Slawomir Lasota. Expspace lower bounds for the simulation preorder between a communication-free petri net and a finite-state system. *Inf. Process. Lett.*, 109(15):850–855, 2009.

12. Anca Muscholl and Igor Walukiewicz. A lower bound on web services composition. *Logical Methods in Computer Science*, 4(2), 2008.